Appendix A

I. Foreword

This document describes the Hypertext Control System (HCS), which is a system for controlling and observing a device using a controller capable of hypertext communication.

II. Scope

Description of Remote Control System for Control of Devices via Hypertext over Proximity Bearers.

III. References

a. GSM for MSISDN
b. Bluetooth for BTi
c. IrDA for Iri
d. RFC for HTTP

IV. Acronyms and abbreviations

BT	Bluetooth
Bti	Bluetooth Interactive
CED	Consumer Electronic Device
DD	Device Domain
DHCI	Device Hypertext Control Interpreter
GSM	Global System Mobile
HC	Hypertext Controller
HCID	Hypertext Controller Identifier
HCL	Hypertext Control Language
HCRF	Hypertext Controller Request Filter
HCVM	Hypertext Control Virtual
HCVM	Hypertext Control Virtual Machine
HEnc	Hypertext Encoder
HIPE-L	Hypertext Interactive Proximity Environment Language
HPS	A Hypertext Protocol Stack
HTTP	Hypertext Transport Protocol
IMode	Internet Mode (Japanese Mobile Internet Standard)
ISO	International Standards Organisation
LC	Link Controller
MSISDN	Mobile Station International Subscriber Digital Number

PB	Proximity Bearer
PLC	Proximity Link Controller
WBXML	Wireless Binary XML
WML	Wireless Markup Language
WSP	Wireless Session Protocol

Introduction

This document describes the Hypertext Control System (HCS), which is a system for controlling and observing a device using a controller capable of hypertext communication. More particular, but not exclusively, the system relates to methods of, computer programs for, and apparatus for control and observation of a device from a mobile communications device via a proximity bearer.

Background

WAP & WML

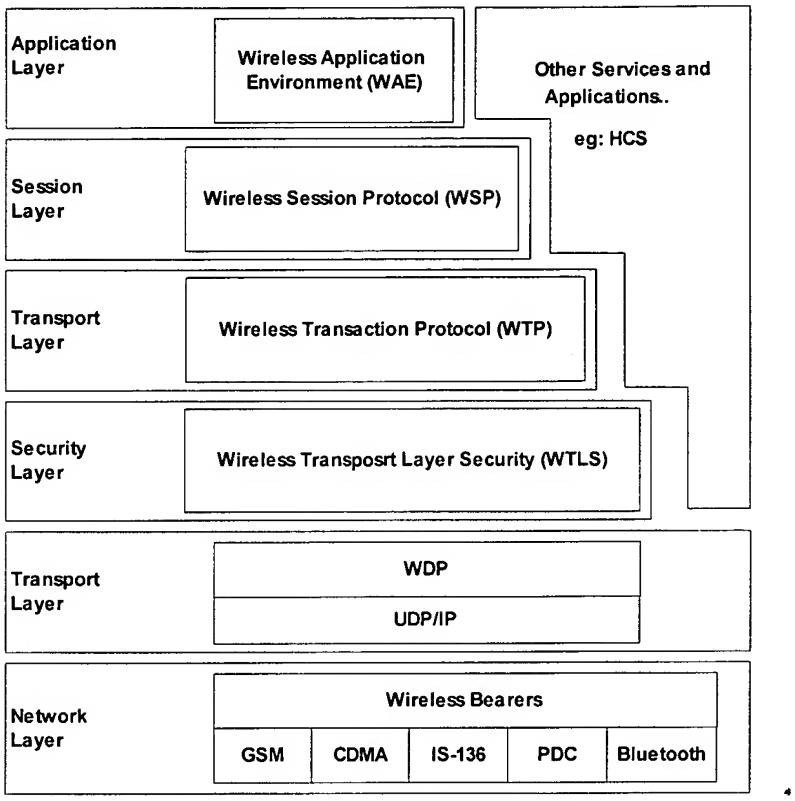
The Wireless Application Protocol (WAP) is a standard for for the presentation and delivery of wireless information and telephony services on mobile phones and other wireless terminals.

To date wireless terminal manufaturers representing 90 per cent of the worlds market across all technologies have joined the WAP Forum™.

The WAP has been developed as much as possible with the use of existing industry standards in particular Internet standards. The WAP forum also has formal liason with International standards and specifications bodies such as:

W3C, TIA, ETSI etc These facts make WAP of particular interest as a vehicle for such control systems such as HCS.

The markup language used in WAE is Wirelwess Markup Language(WML). WML is a tag-based document language specified as an XML document type. Thus, WML may be generated with many standard authoring tools. Dyamic WML can be generated by such standard mechanisims as CGI, Perl ASP and similar. WML coding has been specially designed to make efficient use of Wireless bearers.



The Wireless Application Protocol Stack

Bluetooth

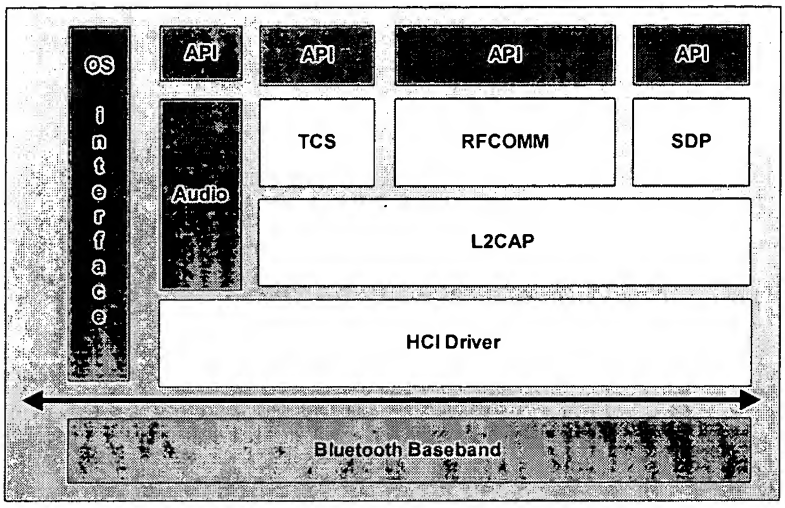
Bluetooth is a short range radio link intended to be a cable replacement between portable and or fixed eletronic devices.

Bluetooth operates in the unlicenced ISM band at 2.4GHz. A frequency hop transceiver is applied to combat interference and fading. The symbol rate is 1Ms/s and a slotted channel is applied with a nominal slot lenght of 625 micro seconds. A TDD multiplex system is used for full duplex operation.Bluetooth can support up to 3 simultaneous voice channels at 64k/bps.

Bluetooth provide point to point and point to multipoint connections.

The Bluetooth Link manager supports inquiry and paging proceedures which allow a Bluetooth module discover which units are in its transmission range. This GIAC general inquiry mode

allows the source bluetooth device to select certain device types via a DIAC (dediacted inquiry access code)



RFCOMM:
This is the transport layer of Bluetooth with provision for RS-232 serial port emulation. This protocol supports up to 60 simultaneous connections between Bluetooth Devices.
Rfcomm can transmit up to 32Kbps over each link.

SDP
This is the Service Discovery Protocol which provides a means for applications to discover which services are available and to determine the characteristics of those available services using an existing L2CAP connection. The service discovery appliation does not make use of the SDP as a means of accessing services but rather as a means of informing the user of a Local Device the services that are available on the Local Device and or via Remote devices.

L2CAP
This is the Logical Link control and application Protocol. This provides connection-oreinted and connectionless services to the upper layers.

HCI Driver
This establishes a link between the hardware and the protocol stack. and is specific to the Baseband implementation.

The I²C-Bus

The i²c bus is a simple bi-directional 2-wire bus for efficient inter-IC communication.

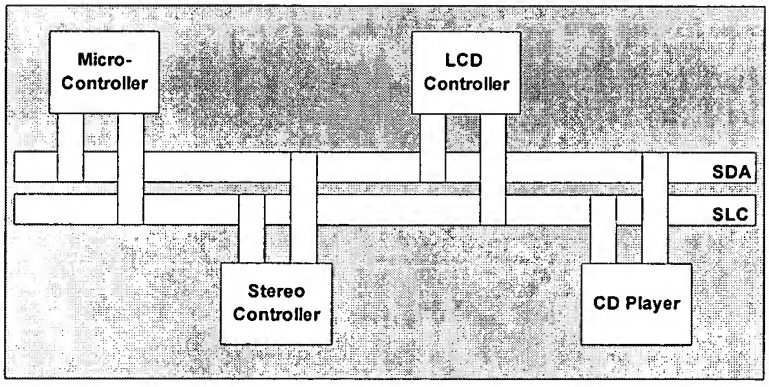
The system allows direct connection of IC controllers for interfaces such as LCD controllers I/O ports and as in the example in this document MMI controllers.

The bus consists of 2 lines

SDA Serial Data Line
SCL Serial Clock Line

Each device connected to the bus has an individual address to allow any device to communicate with ay other device on the bus. Communication is done on a master-slave basis however the bus is a true multi-master bus with collision detection and arbitration between masters.

Serial 8 bit data transfer is supported fro 100Kb/s to 400Kb/s. The supporting interface chips have spike filtering on the data lines and the maximum number of chips that can be connected is only limited by the line maximum capacitance of 400pF.



Blue Velvet

BlueVelvet is a complete Bluetooth system fully integrated in a **single silicon chip**. The chip integrates: RF front end, baseband, ROM, RAM, MCU (ARM7TDMI operating to up to 40MHz) and Peripherals A pure CMOS solution is used to achieve a very low cost solution. The RF-CMOS8 process especially optimized for Bluetooth single chip system is used.

BlueVelvet is compliant with Bluetooth specification V1.0 for Class 2 (0dBm) and for Class I (20dBm) using an external power amplifier.

Chip Features

- 32 KBytes internal SRAM, 200Kbytes internal ROM
- Low power consumption, different power modes and wake-ups
- Two IRDA interfaces
- 8 general purpose PIO pins
- JTAG debug interface and debug signals for logic analyser support
- I2C interface
- Integrated 12Mbit/s USB interface V1.1 in slave mode
- PCM

Linear (13-16bit) or U-law or A-Law (8bit)
Fr: 8KHz,Clock: 200KHz-2MHz

SW Support

ST is providing a set of base standards licensed from Ericsson™ to ensure interoperability:

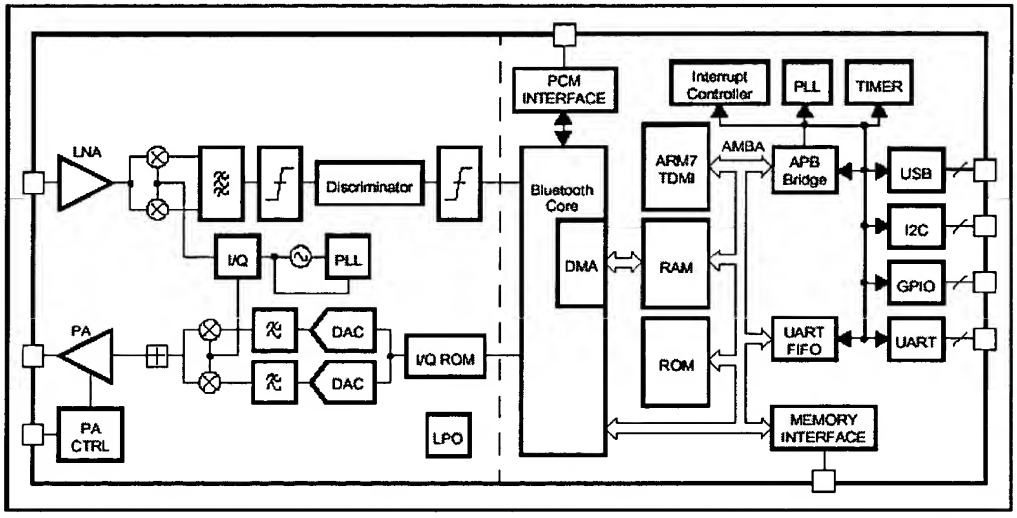
Basic Bluetooth stack

- Baseband, LMP, HCI, L2CAP, SDP
- Higher layer Bluetooth protocols: RFCOMM, TCS-BIN, SDP,...

ST is supporting different software implementations

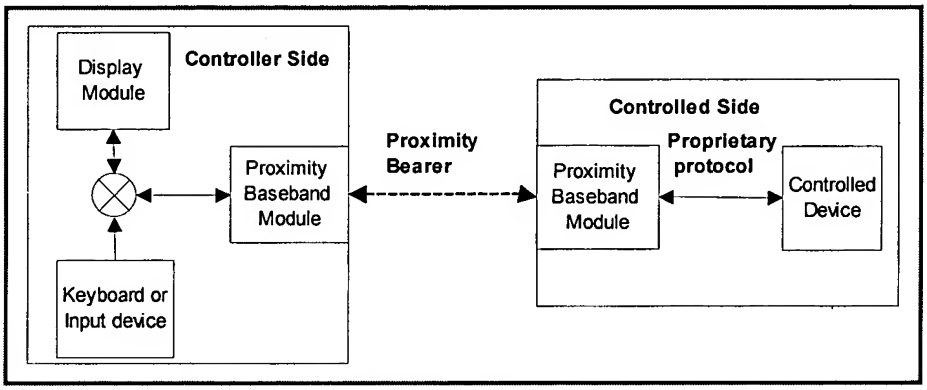
- 2 processors solution based on HCI interface
- 1 processor solution with embedded application

Figure 1 : STw2400 Block Diagram



System Overview

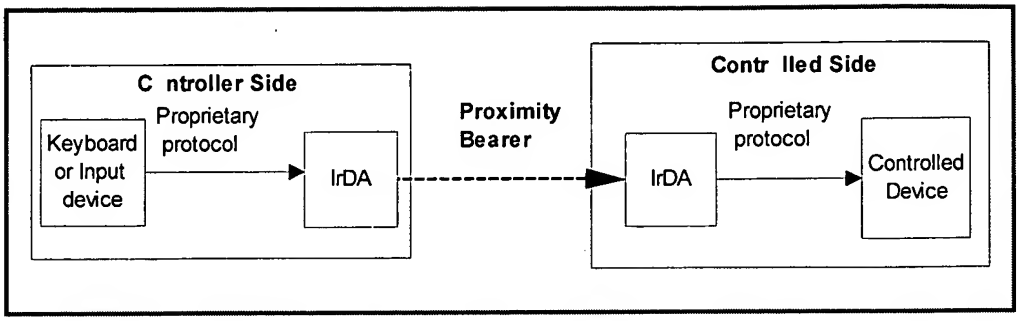
This is a System which allows any device capable of Hypertext communication to control/observe a Device which contains a HCS-Link Controller(HCS-LC). The communication between the Controller and the Controlled Device is typically but not necessarily a Proximity Bearer(PB). In the case of a Proximity bearer this HCS-LC is simply called a Proximity Link Controller(PLC).



Comparison with traditional systems

Implicit in a traditional remote control and observation devices the following assumptions:

1. The Remote Controller has prior knowledge of the command set or protocol used to control the CED device.
2. The Remote Controller has a specific command set for only a specific device or a specific set of devices from a single manufacturer of CEDs.
3. Unreliable link for most IrDA-based controllers – no message acknowledgement from the CED to the remote controller.
4. No security protocol between Remote Controller and CED.



HCS is novel in at least the following ways:

1. The Controlled device makes its presence known to any device wishing to control it by identifying the Link as a HCS controlled Link.
2. There may be master-slave or peer to peer control and observation.
3. The Controlling device needs no prior knowledge of any Controlled device other than its compliance to HCS.
4. The instantaneously available command set of the controlled device is communicated to the controlling device via Hypertext.
5. The control MMI is interactive and is programmed as an interactive State machine within the Bti device generating Hypertext Language in response to the Controller Command and the Device State.
6. The interactive MMI State Machine is written in a Language Called HIPE-L which runs on a link embedded Hypertext Virtual Machine. Command Translation Hypertext generation within the HCI (this is specific to our current implementation- not general)

Main features of the HCS System are:

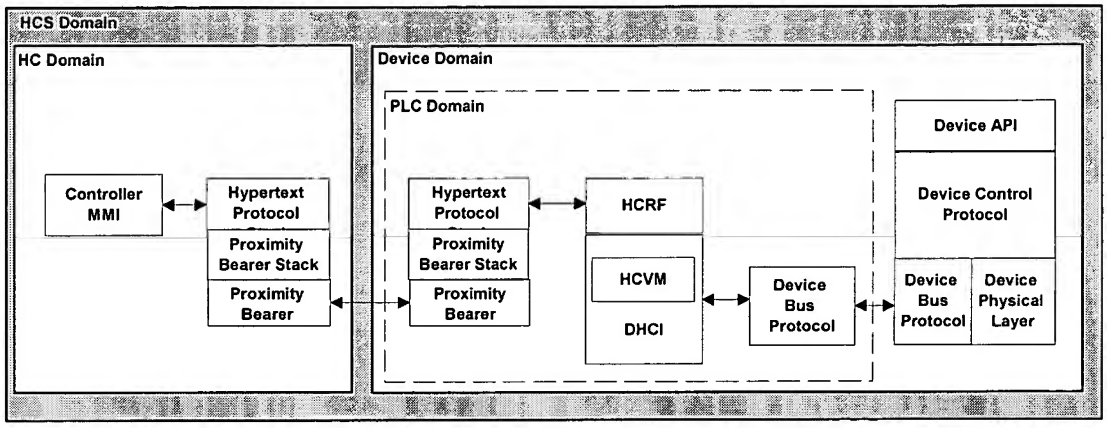
1. Control and observation of a device equipped with a HCS-Link Controller using hypertext communication
2. Controller independence of the Controlled Device, Control Protocols and Command Syntax.
3. HCS-LC discovery and Selection (initiated by prospective controller)
4. HCS-LC Session management
5. HCS-LC Access Rights & Security management

Enhanced Remote Control

Enhanced features of HCS remote control are:

1. Reliable link between HC and CED: usually there is no notion on a remote controller if a command has succeeded or not. The link is typically not reliable.
2. Real-time CED-status-dependent control and observation menu generation. The generated menu on the HC is dependent on the state of the consumer electronic device (CED). For example if a CD is playing, the options for Pause and Stop CD are displayed. If the CD is stopped then Play option is displayed.
3. Control menus downloaded from the CED/PLC to the HC. Not stored on HC long-term.
4. Interactive control involving both HC and CED/PLC
5. More control using this medium. Each device can provide its own complex structure of menus to the HC (controller).
6. Control and observation status and command menus are passed as hypertext information which is rendered on the HC (Remote Controller).
7. Hypertext type independence. The controller can cope with generation of a plurality of hypertext types, for example WML, HTML and cHTML over different bearers.

System Architecture



The diagram above shows the modular system overview of the HCS. We have segmented the system into natural modules and domains, relating to the Hypertext Controller (HC) and the Device Domain (DD). The Device Domain includes the PLC Domain which performs the Hypertext Control and Access, as in most use cases the PLC is situated within the consumer electronic product.

Proximity Bearer (PB)

A Proximity Bearer (PB) may be defined as a wireless communication channel within an area of less than about 100 meters. An example of such a Bearer is Bluetooth where most communication takes place between devices within a 10 meter radius.

The Hypertext Controller (HC)

This is an apparatus capable of hypertext-based control and/or observation of a selected device over a proximity bearer. More particular, but not exclusively, the HC apparatus may be a mobile communications device or a personal digital assistant (PDA). The HC may have a standard Proximity bearer baseband module or indeed a PLC as described in this HCS description.

Proximity Link Controller (PLC)

This is an apparatus used for detection of devices that are operating using the same bearer or bearers, that manages the proximity bearer, physical link, link integrity and networking over the bearer, link status, unique link identification, connection control amongst other functions. The PLC may additionally include the HCRF and Hypertext Control Interpreter (HCI).

Hypertext Controller Request Filter (HCRF)
This is a filter that directs control and observation-related Hypertext Request to the Hypertext Control Interpreter.

Device Hypertext Control Interpreter (DHCI)
This is a computer program that interprets Hypertext Requests using a Hypertext Control Virtual Machine that executes compiled Hypertext Control Language (HIPE) programs.

Hypertext Controller Virtual Machine (HCVM)
This is a virtual Hypertext machine used by the DHCI program to generate the correct Hypertext format and encoding in response to the HC requests. It may also have included in it the format and encoding of the command set for the Device in which the PLC resides*. This allows the programmer of a HCI to issue commands in an abstract way to the Device and issue Hypertext responses independent of the Hypertext type used.

Hypertext Protocol Stack (HPS)
A Hypertext Protocol Stack is a protocol stack based on the ISO model that is required for a particular Hypertext Protocol to be used. For example with a WAP mobile phone, the WAP Server Protocol stack is required on the PLC for WML-based communication and related encoding to take place.

Note * (This functionality may alternately be included in a Command transcoder module if the device manufacturer wished to keep proprietary the format and encoding of the device commands allowing third-party programming of the DHCI)

Functional Specification

The Hypertext Controller (HC)

The HC (typically a mobile communications device or a personal digital assistant (PDA)) provides or has access to module providing the following:

1. Proximity Bearer Protocol Stack
2. A Hypertext Communications stack (eg: WAP)
3. Mechanisms for performing Hypertext Communication via this Proximity bearer
4. Mechanisms for Proximity bearer detection
5. Mechanisms for Proximity bearer Selection for use as a Hypertext medium.

Whereas it is not excluded that the HC-MMI may have HCS specific commands it is not a requirement. Typically all of these requirements are fulfilled by a communications device with Hypertext enabled for communications over a detected proximity bearer.(eg: Mobile phone with a WAP enabled Bluetooth bearer)

Proximity Link Controller (PLC)

Requirements

This is the core of the HCS system and provides the following Functions:

1. Proximity Bearer Protocol Support
2. Hypertext Communications Support (eg:WAP)
3. Mechanisms for performing Hypertext Communication via this Proximity bearer
4. Mechanisms for Proximity bearer detection and presence signaling
5. Mechanisms for Proximity bearer Selection for use as a Hypertext medium.
6. Hypertext Request Filtering (HCRF)
7. Hosting of a Hypertext Virtual Machine (HCVM) for execution of the DHCI module
8. Device Bus Protocol support
9. Device Bus specific Command transcoding (optional)

Functionality

This module is responsible for providing a device with the capability of being controller via a HC

Hypertext Controller Request Filter (HCRF)

Requirements

This module is typically but not escentially a component of the PLC which provides the following functions:

1. Decoding of incoming PLC Hypertext Request Headers
2. Filtering one or more URL requests to the DHCI module for processing
3. Transparent transport of non HCS URL requests.
4. Support of multiple Hypertext formats (optional depending on DHCI implemetation)

Functionality

This module has the task of recognising the returned requests for URL based upon their header information and routing it to the DHCI to allow execution of the related Device command.

The HCRF uses a Uniform Resource Locator data string prefix (up to the full length of the URL data string) or coded version thereof to identify Hypertext Requests to be filtered into the Hypertext Control Interpreter. The URL will include the prefix which identifies its control/observation request purpose and other parts of the URL may optionally include a data string describing the requested control or observation command . Alternatively, the control and observation command data may be stored in other parts of the Hypertext Request, such as HTTP Headers in HTTP protocol. Other protocols that may be used for hypertext communication are but are not limited to WSP protocol.

Hypertext Control Language (HCL)

Requirements

The Hypertext Control Language is required to include the following functionality reagrding its syntax:

1. Multi-lingual output definition
2. Definition URL Links and pictograms to be sent to the HC
3. Device bus Command/Response
4. Timing/Timer definitions

Device Hypertext Control Interpreter (DHCI)

Requirements

This module handles Hypertext Protocol Session as well as configuring the HCVM for execution of required Compiled Code necessary to perform a specific obervation or control or menu display task. This module can be seen as a configuration module for the HCVM. The main required functions are:

1. Invoking and configuration of the HCVM
2. HCVM Response Handling per (Hypertext type, Timeout state, Device state

3. Hypertext Response Dispatching
4. Hypertext Session Management
5. Hypertedxt Request Processing
6. HC and User Profiling
7. HC and User Security

Functionality

The DHCI produces hypertext or encoded version of the hypertext in the same Hypertext syntax and human language as the Hypertext Request that was issued by the Hypertext Controller. The menu language may be specified by a Hypertext Protocol Header such and a HTTP Header in HTTP e.g. ("Accepts") or by a Hypertext Controller Identifier - specific setting stored on the PLC.

The headers that the DHCI requires are the following:

- Accepts: language capabilities and character sets
- User-Agent: browser type and version
- Other Specific iMode WAP or HTML headers: used for identification of the Hypertext Protocol and also for guidelines such as capabilities or screen sizes.

The DHCI takes the form of a program written in a langauage called HIPE. This program is compiled to an intermediate format which is readable by the HCVM.
In this way the different responses in the different natural languages may be pre-compiled. The knowlwdge about hypertext encoding is considered fixed and this is part of the HCVM.

Hypertext Controller Virtual Machine (HCVM)

Requirements

This module is a component of the PLC and is the virtual hypertext machine on which the DHCI code is run. It provides the following functions:

1. URL request dependant State Transition management
2. Timeout management
3. Device State enquiry
4. Hypertext generation & formating
5. Device Command generation & formating*
6. Device Command Dispatching *

Functionality

This module adapts its state dependant on the programing of the DHCI as the DHCI will prime the module and configure the HCVM with the following information:

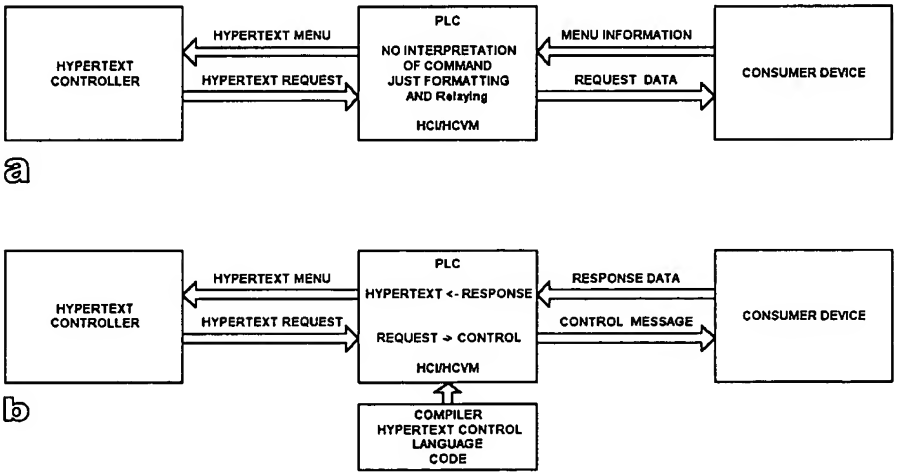
- Hypertext Language currently required (for example WML)
- The parameters that are to be passed top a function as aguments
- The function to be executed
- User Specific and HC specific data, such as HCID

The HCVM will the use the Compiled Code (CC) and execute the required function.

The HCVM will return out of the function producing the required Hypertext as output in standard or encoded form and passing suc Hypertext content to teh DHCI.

Note * (This functionality may alternately be included in a Command transcoder module if the device manufacturer wished to keep proprietary the format and encoding of the device commands allowing third-party programming of the DHCI)

The scenario where HIPE and CC are generating Hypertext responses that assumes that the Consumer Device bus provides only command responses is shown in b). An alternative method is to allow a consumer device to create menu syntax and the PLC DHCI converts such syntax into required hypertext and hyperlinks as shown in a).



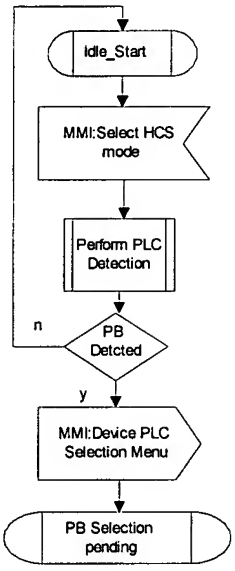
Procedures

Initial BT/PLC Detection Procedure

The HC user selects the option on an MMI menu to enter into "Hypertext over PB" CED detection mode. (Fig X1) . Such a menu option may be denoted by "Find Local Devices" on a phone MMI menu. After detection is completed the user is presented with a list of the available devices to connect to. This list typically contains device descriptive text extracted from the PB-Profile information conveyed upon detection.(Fig X2)

This can be done in two ways:

- A HC Menu (such as a mobile phone) is used to display the list of devices.
- The HC enters a hypertext browser and submits a Hypertext Request to a local PLC in the HC, that has an embedded server function for proximity device selection, to initiate the device enquiry described above and return a list as a Hypertext Response and be able to act upon it.



(Note: MMIs for both selection option 1 and 2 look as figures X1 and X2.)

Filtered Detection (optional)

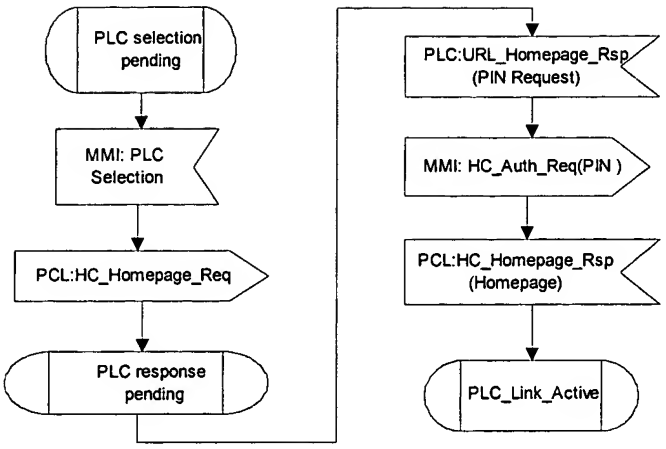
Upon choosing PB Detection the HC shall commence filtered detection of PB Devices. The filtering is done by forming an internal list of all CED devices which responded to the unique PB Device Access Request (URL request) from the HC. The device shall respond to the HC with a PB Device Access Response (figure 2).These responses are recorded and provide CED name identifiers which are presented to the user as a typical HC system menu (option 1) or Hypertext (option 2).

Such menus may optionally be produced with device categorisation according to functionality.

Initial PLC Selection Procedure

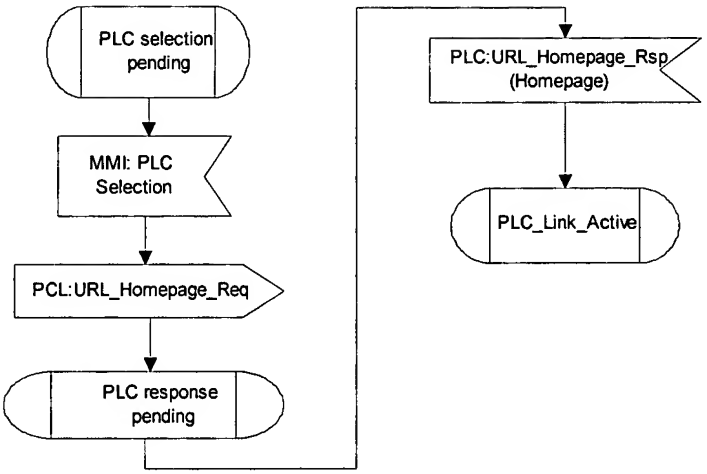
The user now selects from the list the Device Access Response of the Device to be controlled. This issues a request to the Device for access. The PLC looks in the PB Hypertext Controller Device Access Register (HC-DAR) upon reception of every Hypertext command for control or observation and since this is an initial access by the controller device there is no matching entry in the register for the controller device. If the PB is Bluetooth, the Bluetooth Identifier may be used to identify the HC, which is obtained by the DHCI of the PLC from the Proximity Bearer Stack of the PLC.

The non-existence of a matching HCID in this register shall result in the Controller receiving a Hypertext response requesting a device specific Access Code. If correct the device shall (1) register the HCID along with its access classification in the HC-DAR and (2) serve the Control and Observation Deck associated with this access code to the controller device.



Subsequent PLC Selection Procedure

In all subsequent access made by a controller device on a PLC the HCID shall be rercognised as an allowed controller and shall be directly served with Hypertext containing Hyperlinks to Control and Observation functionality.



PLC Configuration Procedure

Global and user-specific settings may be configured optionally after entry of the PLC Maser PIN code. The security aspetcs of such configuration otions are covered in the Security Issues section

Issuing Control & Observations Commands by HC

The Command is issued simply by making the request for a link concerning the control itself, usually by making a key press on a mobile communications device or selecting and pressing an area of a touch screen. Audio and tactile input is allowed for such selection. This Command is encoded as a Hypertext Request and later identified in the PLC by the HCRF and filtered to the DHCI for interpretation.

The Hypertext Request can be any of the common fixed and mobile Internet Hypertext Protocols (WAP/WSP, HTTP, cHTML/HTTP). The Hypertext Response will be of the same type as the Hypertext Request. For example, if the Hypertext Request was a WAP WML request, the response will be in encoded WML.

The HIPE language allows the definition of devices within the language. Devices are defined as bus devices that can be accessed over a specified bus and device identifier. Such busses incude: serial, I^2C, USB, SCSI and others. The language can be used to construct specific functions that will send command strings and receive status strings or acknowledgements back from a identified device on a specific bus.

The HIPE language construct used is as follows:

```
REQUEST          bytes_IDENTIFIER          FROM
dev_IDENTIFIER
   GIVING          bytes_IDENTIFIER          OF
int_IDENTIFIER WITH TIMEOUT
```

For example, the following command from the sample program included in Appendix 1 sends a command to a CD player and waits for a response with timeout:

```
DEVICE cd IS I2C(1)
....
REQUEST output FROM cd GIVING input OF ilen
WITH TIMEOUT 500
```

The DEVICE command defines a specific device.

The `output` parameter is a byte stream that is relayed to the device cd. The `input` parameter is the byte stream that will be received back from the CD player device.

The Hypertext Response is produced by the DHCI. The DHCI configures the HCVM to output a specific type of Hypertext. The HCVM runs Compile Code that has been compile and assembled form source files that have been programmed in a special language called HIPE. The generated hypertext from the HCVM forms the basis of the DHCI Hypertext Response to the HC.

The interpretation of the requests has been programmed in the DHCI code which runs on the HCVM. The HCVM allows the DHCI code to be very compact and to support multiple Hypertext languages and have multi-lingiual or even user selectable human langage responses. Example of a HIPE-L program is descriped in the Language Overview

A Hypertext Encoder may need to be optionally present in system if the DHCI does not produce encoded form of Hypertext for certain Hypertext Language and Protocol combinations, for example iMode and HTTP or WML and WSP. The DHCI may also generate already encoded hypertext, i.e. WBXML-encoded WML format is output directly from the HCVM for WML.

Note: The Compiled Code (CC) is the binary file that is produced by compiling and assembling a source code of a program written in the HIPE Language. The CC is of a specific type as it contains device-specific control and/or observation command definitions for one or more devices. Furthermore, the CC contains control and observation menu definitions defined in one or more human languages – used for internationalization. The CC is intended to be stored in storage accessible to the Blue Velvet Chip. The HCVM the interprets the COMPILED CODE is capable of using menu definitions defined in the CC to generate the appropriate Hypertext Responses to a Hypertext Request that has been requested by from the HC and passed into the HCVM by the DHCI.

Security Issues

User Authentication, Identification and Authorisation

When a user using the HC attempts access to a CED, several methods of user authentication and HC authorisation can take place. The techniques usually involve user, use-level or Master PIN codes:

- User PIN code: is attributed to a particular HC and user
- Use-Level PIN code is attributed to any allowed HC for a particular level of usage of the CED system
- Master PIN Code: is used to reset system or configure other PINs or users or HCs.

The following authentication scenarios have been envisaged:

1. Using a master code to allow/deny use-level PIN usage.
   Use a use level PIN code to access device without subscription of any HC.
2. Using Master PIN code to subscribe a HC and no further authentication of user.
3. Use a Master PIN code to subscribe a HC and each time authenticate user using a user specific PIN.
4. Use a Master PIN code to subscribe a HC and each time authenticate user using a use level specific PIN.

CED-specific Access Classes

There shall be a minimum of 6 BTi device specific classes

Class 0 shall represent a Barred device (optional)
Classes 1..4 Device Bti user codes(1 mandatory)
Class 5 Device BTi master code (mandatory)

The normal entry mode is with one of the 4 the user classes
It is intended that the device manufacturer shall set up a list of all the functionalities and data controllable or displayable by a device and the Master Code shall allow the user to attribute groupings of these functionalities to each of the 4 user classes. In this way closed controller user groups can be supported.

Encryption

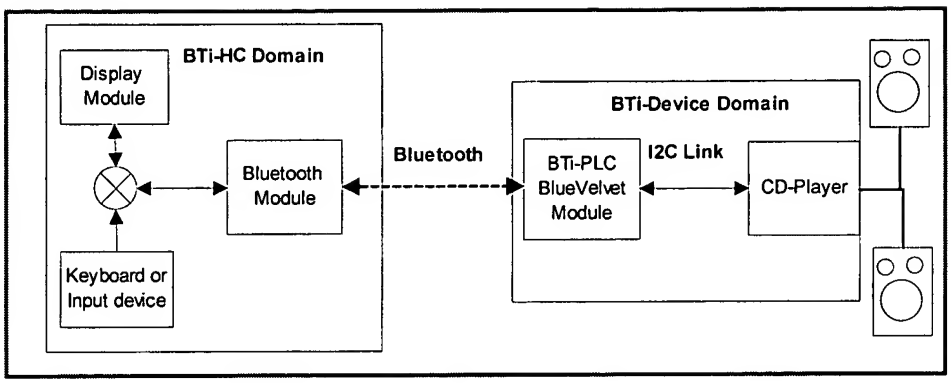
The PB channel should be encrypted for all security sensitive applications such as door access, car access, mobile commerce applications etc.

Implementation Examples

Bluetooth Interactive (BTi)

The BTi system is an implementation of the HCS system using Bluetooth as the Proximity Bearer (PB) and WML as the Hypertext Language. In particular the controlling device is a mobile device (Phone or PDA) which is capable of WAP over Bluetooth. The controlled device is a CD player with an integrated amplifier. The PLC is a Bluetooth silicon module called "Blue Velvet", into which the DHCI has been embedded, although a less integrated solution is also possible. Thus it is possible to communicate between the Mobile device and the audio controller using WML.

BTi Hardware Architecture



BTi-HC Domain

As stated above the Controller is and device which is capable of performing WAP over Bluetooth. No other special requirements exist for the Controller

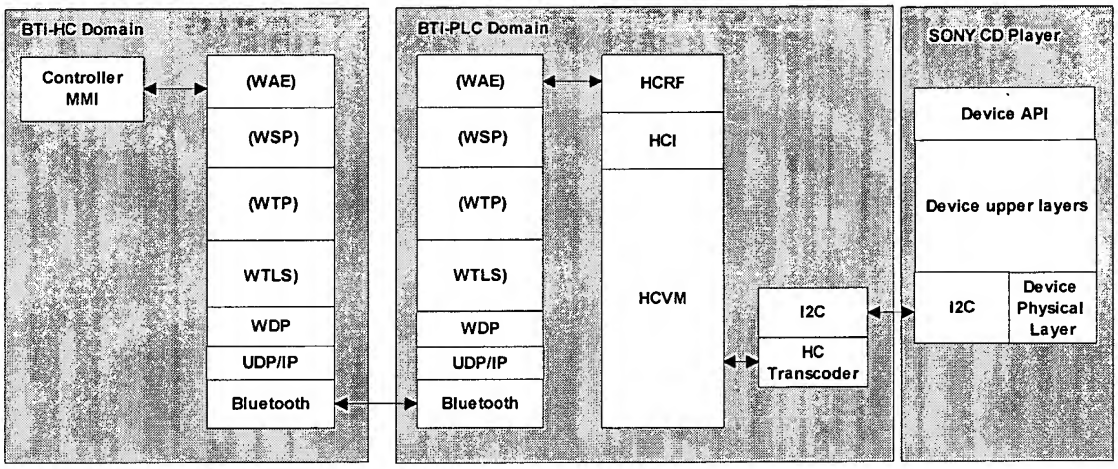
BTi- PLC Domain

The BTi–PLC is a specially adapted version of the STMicroelectronics "Blue Velvet" Chip. The Hardware changes to the Blue Velvet chip depend of the level of integration required but the minimum changes involve an increase of the present internal memory to incorporate the SW required for BTi support and the addition of internal or external "protected" flash for the implementation of the Access Control & Security Registers.

BTi Device Domain

This is represented by a SONY CD Player with and I^2C MMI control interface.

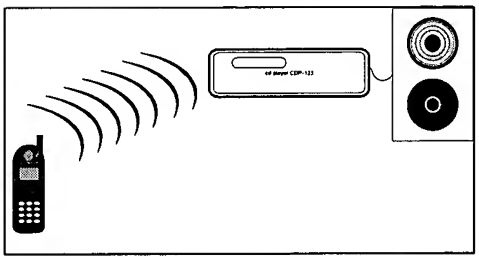
BTi Software Architecture



Functional Scenario for BTi:

CD Player Control using Bluetooth and WAP-enabled mobile phone as HC.

The figure below shows the system overview. A Bluetooth-enabled mobile phone is within 10 meters range of a SONY CDP-123 CD player. The Phone is also WAP-enabled allowing WAP over Bluetooth bearer communication. The user wishes to use the Mobile Phone as an HC (described earlier) in order to control and observe the CD Player actions.

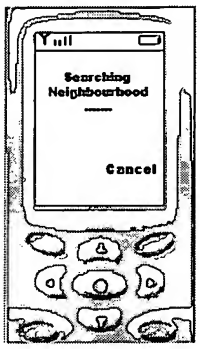


User Activity

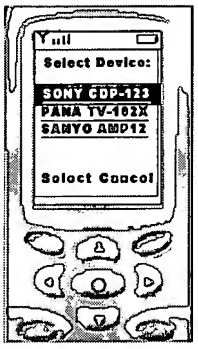
The Mobile Phone scans the Bluetooth Neighborhood as shown in figure X1. Following the Bluetooth enquiry for a number of seconds a list of available devices that are in the Bluetooth neighborhood are displayed as can be seen in figure 2. The user of the mobile phone then selects the desired device (in the example case the user selects the SONY CDP-123 CD player device). Following this device selection a PIN code may optionally be required as shown in figures X7 or X11 requesting the user for either master PIN or user PIN entry. The Security Description is described in the Security MMI Use Case section (below). Alternatively, and on successful PIN code entry the user-specific device-state dependent main menu of the device is displayed as in figure X9 showing a stopped CD player. If the CD is playing then a state-dependent menu is returned as can be seen in figure X10. Figure X10 also shows the observation that the CD player is currently playing track 12. Particular attention is draw to the generation of complex functions such as Select Track in figure X5 where user of the HC can easily select the desired CD tracks to listen to. Following the Selection the user can Play the Disk and only the selected Tracks will be played.

The HC will time out after a period of non-use. The Bluetooth link will be terminated by the HC (master). Subsequent control of the previously selected device will require initial device selection and access procedures as outlined earlier in this section.

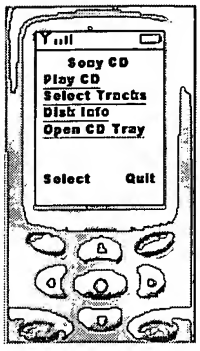
Figures 1-12: Examples of MMI on a mobile phone as HC.



X1

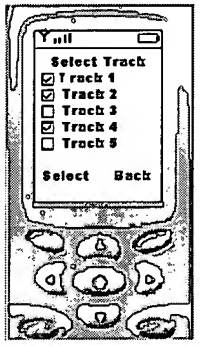


X2

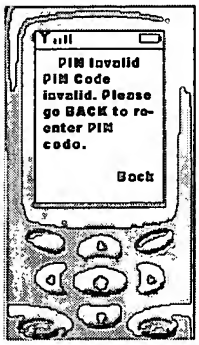


X3

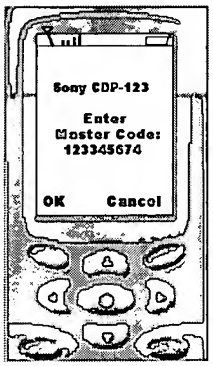
X4



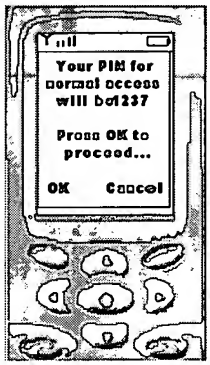
X5



X6



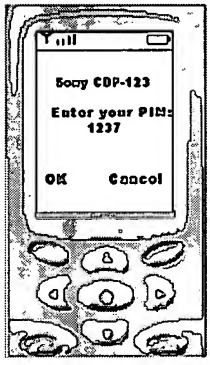
X7



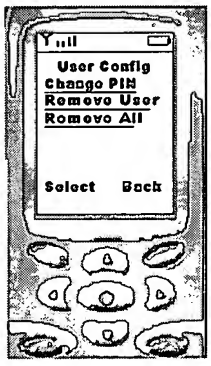
X8

X9

X10



X11



X12

Security MMI Use Case

A Master PIN code may optionally be requested as in Figure X7. A regular user or use-level pin code may optionally be requested as in figure X11. If the user of the HC enter an invalid code as Invalid PIN error warning may be displayed as in figure X6. Repeated incorrect code entry can disable use of the HC with a specific Hypertext Controller Identifier (HCID) for control of the selected CED device.

Note: Hypertext Controller Identifier (HCID) may be a communications protocol identifier attributed to a Hypertext Controller, such as the Bluetooth ID in Bluetooth or the MSISDN in GSM. It may also be a proximity identifier such as a vehicle registration number, were the device requires proximity to the vehicle.

Appendix I: Language Description

HIPE - Hypertext Interactive Protocol Language

HIPE is a new language for interactive control using hypertext languages for user interaction. The language named HIPE-L (Hypertext Interactive Proximity Environment Language) is intended to be used in devices capable of using proximity bearers such as Bluetooth and Wireless LAN (primarily devices capable of proximity ad-hoc networking). However it can be used for any kind of control through Web, WAP or other Hypertext standards.

Example Program in HIPE-L

The example below shows control of basic CD player function using a I^2C-bus-controlled CD player. The program enables the user to control the player in English and Croatian (dependent on jumper settings).

```
1. IMAGE imgbut_play IS "play.jpg"
2. IMAGE imgbut_stop IS "stop.jpg"
3. DEVICE cd IS I2C(1)
4. BOOLEAN valid
5. BYTES input LENGTH 10
6. BYTES output LENGTH 10
7. STRING status LENGTH 20
8. DEFINE JUMPER 0 AS eng WITH main_menu
9. DEFINE JUMPER 1 AS cro WITH main_menu
10.     STRING non_react LENGTH 20
11.     STRING stat LENGTH 20
12.     STRING stpd LENGTH 20
13.     STRING psd LENGTH 20
14.     STRING ply LENGTH 20
15.     STRING cnt LENGTH 20
16.     STRING stp LENGTH 20
17.     STRING paus LENGTH 20
18.     NUMBER num
19.     FUNCTION WILL BE got_track
20.     PAGE WILL BE action_pause
21.     PAGE WILL BE action_stop
```

```
22.     LABEL WILL BE redo
23.     LOCALE GLOBAL [
24.     PAGE action_play [
25.     STORE "G" IN output
26.     SEND output TO cd
27.     DO NOTHING FOR 1000
28.     GOTO LOCALE:redo
29.     ]
30.     PAGE action_continue [
31.     STORE "C" IN output
32.     SEND output TO cd
33.     DO NOTHING FOR 1000
34.     GOTO LOCALE:redo
35.     ]
36.     FUNCTION fetch_status [
37.     TITLE "CD Savvy 123CDP"
38.     STORE 0 IN input(*9)
39.     STORE "S" IN output
40.     NUMBER ilen
41.     REQUEST output FROM cd GIVING input OF ilen
  WITH TIMEOUT 500
42.     IF TIMEOUT OUTPUT non_react ELSE [
43.     OUTPUT stat
44.     NEWPARAGRAPH
45.     IF input(1) = '0' REMEMBER stpd AS status
46.     IF input(1..3) = '255' REMEMBER psd AS
  status
47.     EVALUATE input(1*) GIVING num
48.     IF (num > 0) AND (num < 255) DO
  LOCALE:got_track
49.     OUTPUT status
50.     IF num = 0 BUTTON ply FOR action_play
51.     IF num = 255 BUTTON cnt FOR action_continue
52.     IF (num > 0) AND (num < 255) [
53.     IF WML [
54.     BUTTON paus FOR action_pause
55.     BUTTON stp FOR action_stop
56.     ]
57.     IF HTML [
58.     IMAGEBUTTON imgbut_play FOR action_pause
59.     IMAGEBUTTON imgbut_stop FOR action_stop
60.     ]
61.     ]
```

```
62.     ]
63.     ]
64.     ]
65.     STRING trk LENGTH 10
66.     LOCALE eng [
67.     FUNCTION got_track [
68.     STORE "playing track " IN status
69.     EVALUATE num GIVING trk
70.     CATENATE status WITH trk
71.     ]
72.     PAGE main_menu [
73.     LABEL redo
74.     TITLE "CD Sony 123CDP"
75.     STORE "CD not responding" IN non_react
76.     STORE "Status:" IN stat
77.     STORE "stopped" IN stpd
78.     STORE "paused" IN psd
79.     STORE "Start Play" IN ply
80.     STORE "Continue Play" IN cnt
81.     STORE "Stop" IN stp
82.     STORE "Pause" IN paus
83.     DO fetch_status
84.     ]
85.     ]
86.     LOCALE cro [
87.     FUNCTION got_track [
88.     STORE "svira trak " IN status
89.     EVALUATE num GIVING trk
90.     CATENATE status WITH trk
91.     ]
92.     PAGE main_menu [
93.     LABEL redo
94.     TITLE "Sony CD 123CDP"
95.     STORE "CD ne reagira" IN non_react
96.     STORE "Stanje:" IN stat
97.     STORE "zaustavljen" IN stpd
98.     STORE "stanka" IN psd
99.     STORE "Sviraj" IN ply
100.    STORE "Nastavi" IN cnt
101.    STORE "Zaustavi" IN stp
102.    STORE "Stanka" IN paus
103.    DO fetch_status
104.    ]
```

```
105.    ]
106.    PAGE action pause [
107.    STORE "P" IN output
108.    SEND output TO cd
109.    DO NOTHING FOR 1000
110.    GOTO LOCALE:redo
111.    ]
112.    PAGE action stop [
113.    STORE "X" IN output
114.    SEND output TO cd
115.    DO NOTHING FOR 1000
116.    GOTO LOCALE:redo
117.    ]
```

Multi-hypertext generation

The HIPE-language uses HTTP Protocol Header information, URL request and information relayed in HPS-structured information.

For systems accepting WML, the following lines of HIPE source code:

```
BUTTON paus FOR action pause
BUTTON stp FOR action stop
```

will generate the following code for a WML browser:

```
<p><anchor><go href="url...">Pause</go></anchor></p>
<p><anchor><go href="url...">Stop</go></anchor></p>
```

, and will generate the following source code for a HTML browser:

```
<A HREF="url...">Pause</A><BR>
<A HREF="url...">Stop</A><BR>
```

In a similar fashion the Content-Type Hypertext Response Header is generated and the output from the HCVM is directed by the DHCI to the appropriate HPS. The HCVM handles the multi-Hypertext generation by being configured by the DHCI which receives the HC-originated Hypertext Request.

- END OF TECHNICAL REPORT -